

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2004-005287

(43)Date of publication of application : 08.01.2004

(51)Int.Cl. G06F 15/167
 G06F 9/38
 G06F 12/08
 G06F 15/16
 G06F 15/78
 G06T 1/20
 G06T 7/20
 H04N 7/24

(21)Application number : 2002-160902

(71)Applicant : HITACHI LTD

(22)Date of filing : 03.06.2002

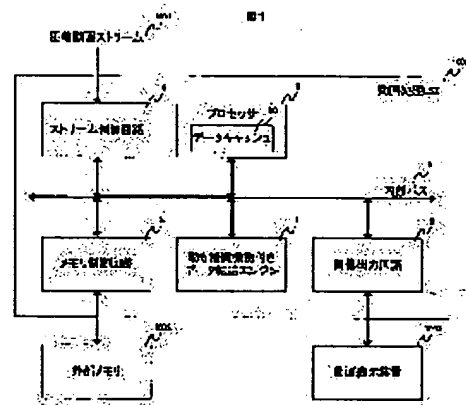
(72)Inventor : TANAKA KAZUHIKO
 HOSOKI KOJI
 TOJIMA SHIGEKI
 NISHIOKA KIYOKAZU

(54) PROCESSOR SYSTEM WITH COPROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To solve a problem on an increase of an area of a chip, found when a memory exclusively used for transferring data is loaded inside of an LSI as a means for transmitting a result of the operation by a coprocessor to a processor, in the LSI provided with the coprocessor besides the processor.

SOLUTION: The data are transferred from the coprocessor to the processor by using a data cache loaded inside of the process. In order to realize this transferring, the data cache is provided with a mechanism for rewriting a tag memory and a data memory in accordance with a writing request from a bus, and the coprocessor is provided with an address generating device for designating the data cache as a writing address.



LEGAL STATUS

[Date of request for examination] 13.05.2004

[Date of sending the examiner's decision of rejection] 15.11.2005

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision 2005-24144
of rejection]

[Date of requesting appeal against examiner's 15.12.2005
decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-5287

(P2004-5287A)

(43) 公開日 平成16年1月8日(2004.1.8)

(51) Int.Cl.⁷

F 1

テーマコード (参考)

G06F 15/167

G06F 15/167

Z

5B005

G06F 9/38

G06F 9/38

370C

5B013

G06F 12/08

G06F 12/08

513

5B045

G06F 15/16

G06F 12/08

551J

5B057

G06F 15/78

G06F 15/16

620G

5B062

審査請求 未請求 請求項の数 15 OL (全 16 頁) 最終頁に続く

(21) 出願番号

特願2002-160902 (P2002-160902)

(22) 出願日

平成14年6月3日(2002.6.3)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(74) 代理人 100075096

弁理士 作田 康夫

(72) 発明者 田中 和彦

神奈川県川崎市麻生区王禅寺1099番地

株式会社日立製作所システム開発研究所
内

(72) 発明者 細木 浩二

神奈川県川崎市麻生区王禅寺1099番地

株式会社日立製作所システム開発研究所
内

最終頁に続く

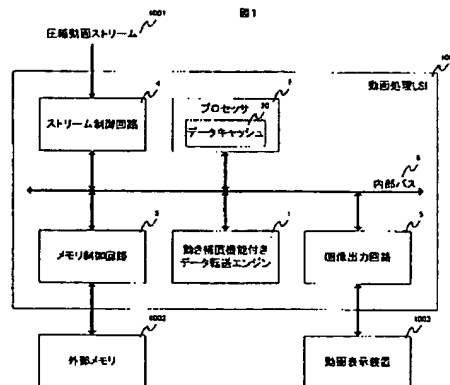
(54) 【発明の名称】 コプロセッサを搭載したプロセッサシステム

(57) 【要約】

【課題】 プロセッサの他にコプロセッサを搭載したLSIで、コプロセッサの演算結果をプロセッサに送る手段として、データ転送専用のメモリをLSIの内部に搭載するとチップの面積が増大してしまう。

【解決手段】 プロセッサの内部に搭載されているデータキャッシュを用いて、コプロセッサからプロセッサへデータを受け渡す。これを実現するために、データキャッシュには、バスからの書き込み要求に応じてタグメモリとデータメモリを書き換える機構を搭載し、コプロセッサにはデータキャッシュを書き込み先として指定可能なアドレス生成装置を搭載する。

【選択図】 図1



【特許請求の範囲】

【請求項1】

キャッシュを有するプロセッサと、
特定の処理を行う第二のプロセッサと、
前記プロセッサ及び前記第二のプロセッサとを接続するバスとを有し、
前記第二のプロセッサは、前記バスに接続されたデバイスに対してアクセスするためのアドレスを生成する手段を有し、
前記第二のプロセッサは、該第二のプロセッサが演算した結果を、前記アドレス生成手段によって生成されたアドレスに基づいて、前記キャッシュに転送することを特徴とするシステム。

10

【請求項2】

前記第二のプロセッサは前記のバスを経由して前記キャッシュにアクセスすることを特徴とする請求項1記載のシステム。

【請求項3】

前記プロセッサと前記第二のプロセッサは異なる命令コード体系で動作することを特徴とする請求項1記載のシステム。

【請求項4】

前記第二のプロセッサから前記キャッシュへの書き込みは、前記キャッシュを構成しているタグメモリに格納されているデータの値にかかわらず実行されることを特徴とする請求項1記載のシステム。

20

【請求項5】

前記第二のプロセッサは動き補償処理を行うことを特徴とした請求項1記載のシステム。

【請求項6】

前記プロセッサ及び前記第二のプロセッサが行った動き補償処理の結果を格納するためのメモリを有し、
前記第二のプロセッサは動き補償処理の途中結果を、前記メモリに格納することなく前記プロセッサへ転送し、
前記プロセッサが前記途中結果を用いて動き補償処理の最終結果を生成し、前記メモリへ格納することを特徴とする請求項5記載のシステム。

【請求項7】

前記プロセッサ及び前記第二のプロセッサはバスブリッジによって接続された異なるバスに各々接続されており、
前記第二のプロセッサは前記異なるバスの各々に接続されたデバイスに対してアクセスするためのアドレスを生成し、
前記第二のプロセッサは、前記バスブリッジを経由して演算結果を前記のバスを経由して前記のキャッシュへ書き込むことを特徴とする請求項1記載のシステム。

30

【請求項8】

前記第二のプロセッサは前記異なるバスのどちらにも接続されていることを特徴とした請求項7記載のシステム。

【請求項9】

前記第二のプロセッサは画像のスケーリング処理を行うためのものであることを特徴とする請求項7記載のシステム。

40

【請求項10】

前記第二のプロセッサは動きベクトルの検出を行うためのものであることを特徴とする請求項7記載のシステム。

【請求項11】

プロセッサが有するキャッシュメモリに対して、前記プロセッサに接続されているバスを経由して演算結果を書き込むために必要となるアドレス生成装置を備えたことを特徴としたプロセッサ。

【請求項12】

50

前記プロセッサ及び前記コプロセッサは異なる命令コード体系を持っていることを特徴とする請求項11記載のコプロセッサ。

【請求項13】

前記演算は動き補償処理であることを特徴とした請求項12記載のコプロセッサ。

【請求項14】

前記キャッシュメモリはデータキャッシュであることを特徴とした請求項1記載のシステム。

【請求項15】

前記キャッシュメモリはデータキャッシュであることを特徴とした請求項13記載のコプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、コプロセッサを搭載したプロセッサシステムに関するものである。

【0002】

【従来の技術】

近年、マルチメディア用のプロセッサを用いたソフトウェアによる動画処理に対する要望が高まっている。動画処理をソフトウェア化することで、新しい規格への即応、LSIの開発経費の低減などが実現できるためである。しかし、動画は静止画や音声に比べて扱うデータの量が非常に大きいため、その処理には膨大な演算量が必要である。

【0003】

動画をソフトウェアで処理する場合、全ての処理を単一のプロセッサで行う技術のほかに、動き補償等、比較的処理が簡単なわりに高い演算性能が要求される処理を特定の処理に特化したプロセッサ（以下「コプロセッサ」）で行い、残りの処理をメインのプロセッサで処理する方法がある。

【0004】

一般に、単一のプロセッサを使用する方法のほうが、チップ面積を抑えられる。しかし、高い演算性能を得るためにプロセッサを高い動作周波数で動作させる必要があり、設計工数や消費電力などの点で問題が生ずる場合がある。

【0005】

一方、コプロセッサを使用する方式では、チップ面積は増大するが、演算負荷が分散されるため、低い動作周波数で動画処理を実現でき、より高い性能が求められる分野や低消費電力化が求められる分野などで有利となる。

【0006】

コプロセッサを使用する方式は、特開2000-251625公報に開示されている。この公報には、コプロセッサの演算結果を、プロセッサとコプロセッサの双方からアクセスできるメモリに一旦格納し、それをプロセッサが読み出すことでプロセッサとコプロセッサの間でデータをやり取りしている。

【0007】

【発明が解決しようとする課題】

前記公報に記載された技術では、プロセッサ、コプロセッサ及びメモリを同一のLSIに集積すれば、これらの間の通信を高速に行うことができる。しかし、この場合には、チップ面積が増大し、システムの製造コストが増大してしまうという問題が生ずる。

【0008】

一方、チップ外部に存在するメモリを用いてデータのやり取りを行うようにすれば、チップ面積の大幅な増大は回避できる。しかし外部メモリに対するアクセスは、LSIの内部動作に比べると低速であるため、システム全体の処理性能が向上しない。

【0009】

チップ面積の増加を抑えて性能向上を実現する技術としては、プロセッサに接続されているキャッシュメモリの代わりに、通常のアドレッシングによりアクセス可能なメモリを搭

10

20

30

40

50

載し、これをプロセッサとコプロセッサの通信にも使用するという技術がある。しかし、この方法はキャッシュメモリを用いずに、通常のアドレッシングによりメモリをアクセスするため、プログラム開発者の負担が増加し、ソフトウェアの開発工数が増えるという問題点がある。

【0010】

【課題を解決するための手段】

本発明では、プロセッサ中のデータキャッシュとコプロセッサをバスにより接続し、コプロセッサが、コプロセッサにおける演算の結果をプロセッサ内部のデータキャッシュへ書き込むという構成とする。この際、データキャッシュのデータメモリへ転送するデータを書き込むだけでなく、タグメモリにも転送するデータのアドレス中の適切なフィールドを書き込む構成としても良い。

10

【0011】

【発明の実施の形態】

図1は、本発明を適用したシステムである動画処理LSI1000の構成を示す図である。動画処理LSI1000は、圧縮された動画を伸張して表示するために用いられる。動画処理LSI1000は、動き補償機能を持ったデータ転送エンジン1、プロセッサ2、メモリ制御回路3、ストリーム制御回路4、画像出力回路5を有し、これらが各々内部バス6で接続されている。

【0012】

プロセッサ2は、内部にデータキャッシュ20を有する。尚、プロセッサ2は命令キャッシュも備えている場合もあるが、本図では省略する。プロセッサ2をシステム中のメインのプロセッサと考えた場合、動き補償機能付きデータ転送エンジン1は、コプロセッサの位置付けとなる。メモリ制御回路3は動画処理LSI1000が外部メモリ1002と通信を行うための回路である。尚、プロセッサ2と動き補償機能付きデータ転送エンジン1で実行される命令体系は、同じでも、異なっても構わない。

20

【0013】

外部メモリ1002は、半導体メモリに限定されず、ハードディスクのように磁気を利用した記憶媒体など、データを記憶できる媒体であれば良い。また、外部メモリ1002は、ネットワークなどを介して離れた場所にあっても構わない。

【0014】

ストリーム制御回路4は、入力された圧縮された動画ストリーム1001をプロセッサ2等へ転送するための制御回路である。メモリ制御回路3とストリーム制御回路4は共通化することも可能である。画像出力回路5は、伸張された画像を画像表示装置1003へ表示するための制御回路である。動画表示装置1003は、ネットワークなどを介して離れた場所にあっても構わない。

30

【0015】

本実施形態では、動き補償機能付きデータ転送エンジン1、プロセッサ2、メモリ制御回路3、ストリーム制御回路4、及び画像出力回路5は全て動画LSI1000へ集積されているとしているが、どの機能をLSIに集積するかについては任意であり、動き補償機能付きデータ転送エンジン1、プロセッサ2、メモリ制御回路3、ストリーム制御回路4、画像出力回路5をそれぞれ一つのLSIで実現して、チップ間バスで接続するといった形態も考えられるし、逆に外部メモリ1002を動画処理LSI1000と同一のチップ内に収めることも可能である。

40

【0016】

動き補償機能付きデータ転送エンジン1は、データ転送を行いながら転送しているデータに対して動き補償に必要な演算を施す回路である。動き補償とは、一般的な動画では連続したフレームのデータは非常に似ていることを利用して高い圧縮率を得る手法であり、MPEG(Motion Picture Experts Group)などで使用されている。

【0017】

50

以下動き補償について簡単に解説する。図2に示すように風景のみが写っている動画では、各フレームの内容はほとんど変化しないため、あるフレームを繰り返し表示することでデータ量を抑えることが可能である。実際には、風景といえども完全に静止しているわけではないので、フレーム間の差分を取り、それを別の方法でさらに圧縮して圧縮後の画像データを作成するといった方法が取られる。

【0018】

例えば、フレーム $n+1$ を圧縮する際には、フレーム n との差分を取り、その差分データを圧縮することで大幅に圧縮率を高めることができる。この場合、フレーム $n+1$ を圧縮対象フレーム、フレーム n を参照フレームと呼ぶ。

【0019】

しかし、この方法は、図3のように動く物体を表示する画像に対しては、圧縮率の効果が低減してしまう。このような場合には、各フレームを小さな領域（以下「マクロブロック」と呼ぶ）に分割してマクロブロック単位に圧縮を行う。この際、参照フレームから圧縮対象のマクロブロックに一番似ているマクロブロックを探し出し、両者の差分を取ることで、データ削減効果を保持することができる。これが動き補償技術である。

【0020】

本技術を使用する場合、差分のデータと一緒に、参照フレーム中のどのマクロブロックを圧縮時に使用したかを示す情報が保存される必要がある。この情報は、一般には、圧縮対象のマクロブロックの座標に対する参照されるマクロブロックの相対座標で表され、動きベクトルと呼ばれる。動きベクトルは、圧縮された動画を伸張する際に必要となる情報の一つである。

【0021】

通常、画像を扱う際の最小単位は画素であるが、実際の動画は画素単位に動くわけではない。このため、動きベクトルの設定値を1画素単位よりもさらに細かく指定できるようにすることでさらに圧縮率を上げることができる。MP E Gでは、ハーフヘルと呼ぶ手法により、動きベクトルの設定値を $1/2$ 画素単位で設定することが可能である。

【0022】

ハーフヘルを使用する場合、参照フレームに対して隣接した2画素または4画素の算術平均による補完処理を施すことで、参照対象のマクロブロックが作成される。また、MP E Gでは、複数の参照フレームを使用して動き補償を行う双方向予測といった技術も使用されている。これらに関しては、MP E Gの規格書および「最新MP E G教科書、藤原洋監修、アスキー出版局」などの文献に詳しく記載されている。

【0023】

動き補償機能付きデータ転送エンジン1は、内部バス6と接続されている装置間のデータ転送を制御することができる。ここで内部バス6に接続されている装置には、データキャッシュ20も含まれる。例えば、メモリ制御回路3を経由して外部メモリ1002からデータを読み出して、データキャッシュ20へ書き込むといった処理を制御できる。この際、動き補償機能付きデータ転送エンジンが、データに演算処理を施すこともできる。

【0024】

図4は、動き補償機能付きデータ転送エンジン1の内部構成示す図である。動き補償機能付きデータ転送エンジン1は、内部バス制御回路301、読み込みバスアドレス生成回路302、バッファ303、バッファ304、データ転送エンジン制御回路305、ハーフヘル処理回路306、307、双方向予測処理回路308、書き出しバスアドレス生成回路309、及び演算結果出力回路310を有する。

【0025】

内部バス制御回路301は、動き補償機能付きデータ転送エンジン1と内部バス6との間のデータのやり取りを制御する。読み込みバスアドレス生成回路302は、データ転送エンジン1が外部メモリ1002などに内部バス6を経由して読み出しアクセスを行う際のアドレス生成を行う。読み込みバスアドレス生成回路302は、バッファ303およびバッファ304に対するアドレス生成も行う。

10

20

30

40

50

【0026】

バッファ303及び304は、内部バス6経由で読み出したデータの格納に使用される。本実施形態では、バッファメモリを2つとしているが、これは2つに限定されたものではなく、1つにすることも可能であるし、3つ以上にすることも可能である。

【0027】

本実施形態では、単純な双方向の動き補償に対応するために2つのバッファを用意する。すなわち双方向予測に使用する一方のフレームのマクロブロックがバッファ0へ格納され、もう一方のフレームのマクロブロックがバッファ1へ格納される。各バッファの容量は処理性能とチップ面積などのトレードオフにより決定される。本実施形態では、縦横共にマクロブロックの大きさより1画素大きいブロックが格納できるだけの容量を想定している。 10

【0028】

仮にマクロブロックの大きさを縦16画素、横16画素とし、1画素あたりのビット数を8ビットとすれば、必要なメモリ容量は、 $(16+1) \times (16+1) \times 8$ ビットとなる。なお、一般的には、画像データはRGBのように複数の成分で表されるが、ここでは説明のため1成分の処理のみに着目する。成分が増えた場合でも、必要な回路を複数個格納すれば簡単に対応できる。

【0029】

ハーフヘル処理回路306、307は、動きベクトルの値が整数ではなく、 $1/2$ 画素の倍数である場合に、画素間の補完を行う回路である。例えば、 x 座標が $n+0.5$ (n は整数)の画素の値は、その両隣の x 座標が n の画素と x 座標が $n+1$ の画素の値を加算して2で除算することにより求められる。本実施形態では、ハーフヘル処理回路306、307の出力画像の大きさはマクロブロックの大きさと同じ 16×16 画素となる。 20

【0030】

ハーフヘル処理回路306、307の出力は、双方向予測処理回路308へ入力される。この回路はMPEGなどの双方向予測の処理に対応する。本回路は、入力された2つのマクロブロックについて画素ごとに平均値を算出し、最終的な参照マクロブロックを生成する。

【0031】

生成された参照マクロブロックは、演算結果出力回路310および内部バス制御回路301を経由して内部バス6へ出力される。出力先のアドレスは、書き出しバスアドレス生成回路309により生成される。出力先の装置としては、内部バスに接続されている任意の装置を指定することができる。 30

【0032】

データ転送エンジン制御回路305は、動き補償機能付きデータ転送エンジン1内部の各ブロックを制御するための回路であり、バッファ303、304のリードライトのタイミング信号等を生成する。

【0033】

なお、ここで説明したハーフヘル処理回路306、307などの演算器やその他の構成要素は一例であり、他の種類の演算器を使用することや、各演算器の個数などを変更することも可能である。 40

【0034】

図5は、データキャッシュ20の内部構成を示す図である。データキャッシュ20は、プロセッサ2の内部にあるプロセッサ内部バス21と内部バス6の両方に接続されている。データキャッシュ20は、アドレス用データバス200、コントローラ201、データ用データバス202、セレクトラ203、204、タグメモリ205、データメモリ206、内部バス制御回路207、及びプロセッサ内部バス制御回路208を有する。

【0035】

コントローラ201は、データキャッシュ20の各ブロックを制御する。タグメモリ205には、データメモリ206の対応するエントリのタグアドレスおよび有効ビットが格納 50

される。データメモリ206には、データが格納される。セレクタ203はタグメモリ205への入力を選択する。セレクタ204はデータメモリ206への入力を選択する。内部バス制御回路207は、データキャッシュ20と内部バス6との間のデータのやり取りを制御する。内部バス制御回路208は、データキャッシュ20とプロセッサ内部バス21との間のデータのやり取りを制御する。

【0036】

プロセッサ2の内部演算器からプロセッサ内部バス21を経由してデータキャッシュ20へデータの書き込み要求が来た場合には、プロセッサ内部バス制御回路208、アドレスデータバス200、及びデータ用データバス202などを経由し、ライトアドレスの一部がタグメモリ205へ、ライトデータがデータメモリ206へ書き込まれる。

10

【0037】

逆に内部バス6からデータキャッシュ20へデータの書き込み要求が来た場合には、内部バス制御回路207などを経由し、ライトアドレスの一部がタグメモリ205へ、ライトデータがデータメモリ206へ書き込まれる。どちらの場合も書き込まれたエントリの有効ビットは1にセットされ、次にプロセッサ2の内部演算器からそのエントリへアクセス要求が来たときにはデータキャッシュ20はキャッシュヒットすることとなる。

【0038】

ここで、動き補償機能付きデータ転送エンジン1がデータキャッシュ20へ演算結果を書き込む方法について説明する。本実施例における内部バス6上の信号の変化の一例を図10に示す。

20

尚、ここでは、説明を簡単にするために、必要最小限の機能のバスを用いているが、スプリット転送対応やバースト転送対応などの技術を用いることで、さらにデータの転送効率を高めることも可能である。また本実施形態では、内部バス6のアービトレーションを行う回路209は内部バス制御回路207にあることを想定しているが、バスに接続されている他のデバイスに内蔵させることも可能である。

【0039】

基準クロック601は、内部バス6に接続された全ての装置に供給される。リクエスト信号602は、動き補償機能付きデータ転送エンジン1が出力し、内部バス制御回路207内にあるバスのアービトレーション回路209が受け取る信号である。動き補償機能付きデータ転送エンジン1は、内部バス6に接続された装置に対してデータを転送する必要が生じたときに、リクエスト信号601として、1を示す信号を出力する。アービトレーション回路209には、バスに接続された全装置からのリクエスト信号が通知される。アービトレーション回路209は、これらの信号の内容をもとにどの装置にバスの使用を許可するかを決定する。

30

【0040】

アービトレーション回路209は、バスの使用を許可する装置に対して、その装置に対するグラント信号を1にすることでバスの使用許可を通知する。本図では、動き補償機能付きデータ転送エンジン1に対し、グラント信号603を1にすることでバスの使用を許可している。

【0041】

動き補償機能付きデータ転送エンジン1は、バスの使用を許可されると、次のサイクルで、書き出しバスアドレス生成回路309が生成したアドレス614を、アドレス信号604として出力する。本実施形態ではアドレス信号604は32ビット幅の信号としており、その上位4ビットでアクセス対象の装置が決定される。

40

【0042】

例えば、アドレス信号604の上位4ビットが「0000」～「0111」ならばメモリ制御回路3、「1000」ならばデータキャッシュ20といった具合に、アクセス対象となる装置が対応付けられる。ここでは、書き出しバスアドレス生成回路309がアドレス614の上位4ビットを「1000」に設定することで、データの書き込み先をデータキャッシュ20に指定している。

50

【0043】

内部バス制御回路301は、書き出しバスアドレス生成回路309が生成したアドレス614をアドレス信号604として出力するとともに、リードライト指定信号605を0に設定して現在のアクセスがライトアドレスであることを指定する。

【0044】

次のサイクルで、内部バス制御回路301は演算結果出力回路310から送られてきたデータ616をデータ信号606として出力する。本実施例では、データ信号606は64ビット幅としている。

一方、データキャッシュ20内部にある内部バス制御回路207は、グラント信号608が1になったサイクルの次のサイクルで送られてくるアドレス614の上位4ビットの内容を調べ、これが「1000」であることとリードライト信号605が0であることから、次のサイクルで送られてくるデータ616はデータキャッシュ20へ書き込むべきデータであると判断する。

【0045】

ここで、本実施形態では、データキャッシュ20は8キロバイトの容量を持つダイレクトマップのキャッシュであり、タグメモリ205は幅20ビット、ライン数1024のメモリ、データメモリは幅64ビット、ライン数1024のメモリで構成されているものとする。1キャッシュラインの容量は、64ビットである。タグメモリには、タグアドレスの31～13ビット目と有効ビットが格納される。

【0046】

アドレス614が送られてくると、アドレス用データバス200は、アドレス614の12～8ビット目を用いて、書き込み対象となるキャッシュラインを決定する。コントローラ201は、ここで決定されたライン番号を用いて、タグメモリ205の対応するキャッシュラインに格納されているデータを読み出す。

【0047】

読み出されたデータに含まれている有効ビットが0である場合には、そのキャッシュラインに対応するキャッシュメモリには、有効なデータが存在しないことを示す。この場合には、コントローラ201は、タグメモリ205の対応するキャッシュラインのアドレス格納部分にアドレス614の31～13ビットを、有効ビット部分に1を書き込み、データメモリ206の対応するラインへデータ616を書き込む。このようにすることで、フロッツァ2がアドレス614に格納されているデータにアクセスする際には、データキャッシュ20がヒットすることになる。

【0048】

タグメモリ205から読み出されたデータに含まれている有効ビットが1である場合には、そのラインには有効なデータがすでに存在していることを示す。この場合には、コントローラ201は、データメモリ206の該当ラインに格納されているデータを、データ用データバス202の内部にあるレジスタへ一旦コピーした後、タグメモリ205の対応するラインのアドレス格納部分にアドレス614の31～13ビットを、有効ビット部分に1を書き込み、データメモリ206の対応するラインへ、次のサイクルで送られてくるデータ616を書き込む。データ用データバス202の内部にあるレジスタへコピーされたデータは、内部バス6が使用可能となった時点で、内部バス6を経由して外部メモリ1002へ書き込まれる。

【0049】

以上、ダイレクトマップ型のデータキャッシュを例に動作を説明したが、セットアソシアティブ型のキャッシュでも本発明の方式を使用することが可能である。この際には、書き込み対象となるラインは、アドレス614の他に、データキャッシュ20の中にあるLRU制御回路などの情報を用いて決定される。

【0050】

次に本構成における動画の伸張処理の流れについて説明する。

まず、最初に動き補償機能付きデータ転送エンジン1を使用しない場合の処理について説

10

20

30

40

50

明する。

【0051】

MPEGなどのアルゴリズムを用いて圧縮された圧縮動画ストリーム1001は、ストリーム制御回路4へ入力される。ストリーム制御回路4は、入力された圧縮動画ストリームのデータを、内部バス6を経由して外部メモリ1002へと書き込む。これと並行して、プロセッサ2は、外部メモリ1002に格納されている圧縮画像データを読み出して伸張処理を行い、その結果得られた画像（以下「原画像」）を外部メモリ1002へ書き戻す。

【0052】

画像出力回路5は、外部メモリ1002から伸張された原画像のデータを読み出し、動画表示装置1003へ出力する。これらの処理は並列に実行できる。通常、このなかで最も処理時間が長いのは、プロセッサ2が外部メモリ1002から圧縮データを読み出し伸張して外部メモリ1002へ書き戻す処理である。すなわち、この処理を高速化することでシステム全体を高速化することが可能である。

【0053】

以下、従来技術における圧縮データの読み出し伸張処理について説明する。

MPEGなど一般に使用されている動画圧縮手法では、圧縮後のデータ列は可変長符号（Variable Length Code）となっている。そこで、まずプロセッサ2は、外部メモリ1002に格納された圧縮ストリームを読み出して、可変長符号の復号処理を行う。圧縮されているストリームを復号することで、動きベクトルおよび離散コサイン変換された画像データ（IDCTデータ）を取り出すことができる。この処理がVLC復号処理である。ここでは、プロセッサ2がこの処理を行っているが、VLC復号処理専用のコプロセッサを搭載することも可能である。

【0054】

この後、プロセッサ2は、IDCTデータに対して逆離散コサイン変換等（IDCT）を施し、動き補償前の画像データを得る。この処理をIDCT処理と称する。この動き補償前の画像データに、必要に応じて参照データを加算することで原画像を得ることができる。

【0055】

次に、プロセッサ2は、外部メモリ1002から参照フレーム中のマクロブロックのデータを読み出し、ハーフペル予測のための画素補完、双方向予測のために複数のマクロブロックの平均化などを行い、参照データを作成する。これを以下「参照データ読み出し+演算」処理と称する。

【0056】

次に、プロセッサ2は、ここで得られた参照データをIDCT処理802の結果と加算することで原画像を得る。これを「加算処理」と称する。最後に、プロセッサ2は、この結果を外部メモリ1002へ書き込む。この処理を「ストア」と称する。

【0057】

本実施形態では、「参照データ読み出し+演算」処理を動き補償機能付きデータ転送エンジン1で実行する構成とする。これにより、他の処理との並列実行を可能とし全体の処理時間を短縮する。

【0058】

図6は、動き補償機能付きデータ転送エンジン1を使用した場合のデータ伸張処理の詳細を示す図である。

本図において、上半分はプロセッサ2が行う処理の内容を、下半分は動き補償機能付きデータ転送エンジン1が行う処理の内容を表している。また、横方向は時間の経過を示している。

【0059】

マクロブロック $n+1$ に着目すると、まずプロセッサ2は、VLC復号処理801b及びIDCT処理802bを行う。次に、プロセッサ2は、VLC復号処理801bで得られ

た動きベクトルを、内部バス6を介して、動き補償機能付きデータ転送エンジン1内部のレジスタへ書き込む。その後、動き補償機能付きデータ転送エンジン1へ、「参照データ読み出し+演算」処理803bを起動する要求を発行する。本実施形態では、この起動操作も、動き補償機能付きデータ転送エンジン1内部のレジスタへプロセッサ2がデータを書き込むことにより行われる。これら一連の処理を完了したプロセッサ2は、マクロブロックnに対する処理を開始する。

【0060】

一方、起動要求を受信した動き補償機能付きデータ転送エンジン1は、自身が有するレジスタに書き込まれている、次に処理するマクロブロックの座標を示す値とプロセッサ2によって書き込まれた動きベクトルの値から、読み込むべき参照マクロブロックが格納されている外部メモリ1002のアドレスを算出する。

10

【0061】

次に、動き補償機能付きデータ転送エンジン1は、算出したアドレスを用いて、外部メモリ1002から必要となるマクロブロックを読み出し、バッファ303へ書き込む。双方向予測を行うことが指定されている場合には、プロセッサ2によって既に書き込まれているもう一つの動きベクトルを用いて、第二の参照フレーム中の第二の参照マクロブロックを読み出し、バッファ304へ書き込む。

【0062】

次に、ハーフヘル処理回路306は、バッファ303の内容を、ハーフヘル処理回路307はバッファ304の内容を用いて、ハーフヘル演算を行う。補完の方法は、それぞれのバッファに読み込む際に使用した動きベクトルの値によって決定される。

20

【0063】

次に双方向予測処理回路308は、ハーフヘル処理回路306とハーフヘル処理回路307からの出力データに対して画素ごとに平均値を求め、双方向予測処理回路308の出力データとする。なお、双方向予測をしない場合には、この平均化処理は不要であり、ハーフヘル処理回路306とハーフヘル処理回路307の出力のうち有効な方を双方向予測処理回路308の出力とする。

【0064】

この双方向予測処理回路308の出力を、データ転送エンジン1は、演算結果出力回路310、内部バス制御回路301、内部バス6を経由してデータキャッシュ20へ書き込む。ここまでの処理が「参照データ読み出し+演算」処理803bである。

30

【0065】

処理803bが終了した後、プロセッサ2は、IDCT処理802bの結果と、処理803bの結果を画素ごとに加算する処理804bを行う。このときには、どちらの結果もすでにデータキャッシュ20へ格納されているため、この処理はキャッシュミスを引き起こさずに高速に行うことが可能である。

最後に、プロセッサ2は、処理804bの結果をストア処理805bにより、外部メモリ1002へ書き込む。

【0066】

このように、「参照データ読み出し+演算」処理803bが、プロセッサ2ではなく、動き補償機能付きデータ転送エンジン1で実行されることで、この処理が実行されている間は、プロセッサ2は他の処理を実行することが可能。本実施形態では、動き補償機能付きデータ転送エンジン1が処理803bを実行している間に、プロセッサ2は、直前のマクロブロックであるマクロブロックnの加算処理804a、ストア処理805aおよび直後のマクロブロックであるマクロブロックn+2のVLC復号処理801c、及びIDCT処理802cを実行する。

40

このように、圧縮データの伸張処理をパイプライン化することで、プロセッサ2の処理能力を無駄にすることなくシステム全体の処理能力を高めることができる。

【0067】

図7は、本発明の第2の実施形態の構成を示す図である。先の実施形態と異なる点は、動

50

画処理LSI1000αは複数の内部バス61及び62を有し、これらバス間がバスブリッジ30で接続されている点である。この例では内部バス6を2つに分割しているが、内部バスは、3つ以上であっても構わない。

【0068】

このようにバスを分割することで、それぞれのバスに接続される装置の数を減らすことができる。さらに、このようにバスを分割することで、バスの物理的な配線長も短くすることができ、動作周波数の高速化または省電力化に有利となる。

【0069】

図8は、図7をさらに発展させた第3の実施形態を示す図である。第一及び第二の実施形態と異なる点は、バスブリッジと動き補償機能付きデータ転送エンジンとを一つの装置とした点である。したがって、本実施形態では、第2の実施形態と同じく、バスを分割したことによる動作周波数の向上などの効果が得られる。さらに、本実施形態では、動き補償機能付きデータ転送エンジン31が二つの内部バスに接続されていることにより、それぞれのバスに対して独立にアクセスすることが可能となる。

【0070】

すなわち、外部メモリ1002から内部バス62を経由して参照データを動き補償機能付きデータ転送エンジン31へ読み込むと同時に、動き補償を行った結果を、内部バス61を経由してデータキャッシュ20へ書き込むことが可能となる。このため、各内部バスのデータ転送負荷を分散することができ、システム全体としてのデータ転送性能を高めることが可能となる。

【0071】

本実施形態で使用される動き補償機能付きデータ転送エンジン31は、内部バス制御回路301に複数の内部バス61及び62が接続される点で、第一の実施形態で使用される動き補償機能付きデータ転送エンジン1とは異なる。

【0072】

また、第4の実施形態として、動画処理LSI1000が有する動き補償機能付きデータ転送エンジン1の代わりにスケーリング機能付きデータ転送エンジン60が搭載される構成がある。

【0073】

図9は、スケーリング機能付きデータ転送エンジン60の内部構成を示す図である。スケーリング機能付きデータ転送エンジン60は、内部バス制御回路601、読み込みバスアドレス生成回路、バッファ603、データ転送エンジン制御回路605、スケーリング処理回路606、書き出しバスアドレス生成回路、及び演算結果出力回路610を有する。

【0074】

スケーリング処理回路606は、表示領域のある範囲に存在する画素に対して2次元のフィルタリング処理を行う回路である。この回路を用いることで、画像処理LSIは、画像の拡大縮小などを行うことが可能となる。また、2次元フィルタの係数を工夫すれば、輪郭強調などの処理を行わせることもできる。このスケーリング機能付きデータ転送エンジン60を用いると、外部メモリ1002に格納されている画像を縮小してからデータキャッシュ20に転送するといったことが可能となる。このようにすると、最終的に要求される画像の解像度が低い場合などは、プロセッサ2に必要とされる演算量を少なくすることができ、

【0075】

また、動画の圧縮処理を行う場合には、動き補償機能付きデータ転送エンジン1の代わりに、動き予測機能付きデータ転送エンジン50を使用することでシステムの性能を高めることができる。動き予測機能付きデータ転送エンジン50は、圧縮対象となっているマクロブロックと最も似たマクロブロックを参照フレーム中から探し出し、その相対座標、すなわち動きベクトルを算出する。この動きベクトルをデータキャッシュ20へ転送する。

【0076】

なお、上述した実施形態では、一つのコプロセッサを搭載するシステムについて説明した

10

20

30

40

50

。しかし、本発明ではコプロセッサとデータキャッシュはバスにより接続されているので、コプロセッサの数を複数にすることや、データキャッシュの数を複数にすることも可能である。コプロセッサの数を増やした場合には、どのコプロセッサからもデータキャッシュに対してデータを書き込むことができる。

【0077】

また、データキャッシュの数を増やした場合には、コプロセッサはデータをどれか一つのデータキャッシュを選択して書き込むことも可能であるし、複数のデータキャッシュに対して書き込むことも可能である。

【0078】

また本発明では、データキャッシュはバスに対して接続されているので、バス上に接続されているコプロセッサの個数を増やした場合でもデータキャッシュの変更はほとんど行う必要がないという利点もある。

【0079】-

また、以上の説明ではデータキャッシュを例に説明したが、本発明はデータキャッシュに特定されるものではない。コプロセッサとしてプロセッサが実行する命令の変換回路などを使用する場合には、その演算結果は命令キャッシュへ書き込まれることになる。

【0080】

【発明の効果】

データキャッシュ方式の持つプログラミングのしやすさという長所を生かしながらコプロセッサの演算結果を高速にプロセッサへ送ることが可能となる。これによりシステム全体の性能を高めることが可能である。また、コプロセッサの数を増やした場合でも、データキャッシュ側の面積の増加を抑えることが可能である。

【図面の簡単な説明】

【図1】本発明を使用したシステムの構成図である

【図2】動き補償を説明する図である。

【図3】動き補償を説明する図である。

【図4】動き補償機能付きデータ転送エンジンの構成図である。

【図5】データキャッシュの構成図である。

【図6】パイプライン化した動画伸張処理の例を示す図である。

【図7】本発明を使用したシステムの別の構成を示す図である。

【図8】本発明を使用したシステムの別の構成を示す図である。

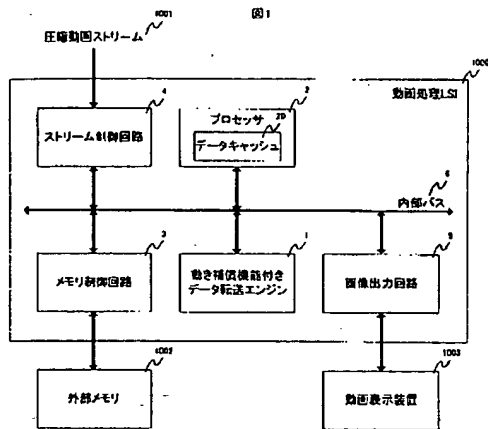
【図9】スケーリング機能付きデータ転送エンジンの構成を示す図である。

【図10】内部バス上の信号の変化を示す図である。

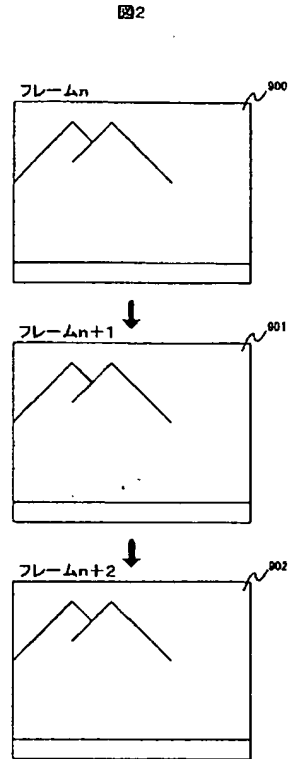
【符号の説明】

1 動き補償機能付きデータ転送エンジン、2 プロセッサ、3 メモリ制御回路、4 ストリーム制御回路、5 画像出力回路、6 内部バス、20 データキャッシュ、21 プロセッサ内部バス、201 コントローラ、202 データ用データバス、203 セレクタ、205 タグメモリ、206 データメモリ、207 内部バス制御回路、208 プロセッサ内部バス制御回路、209 内部バスアービトレーション回路。

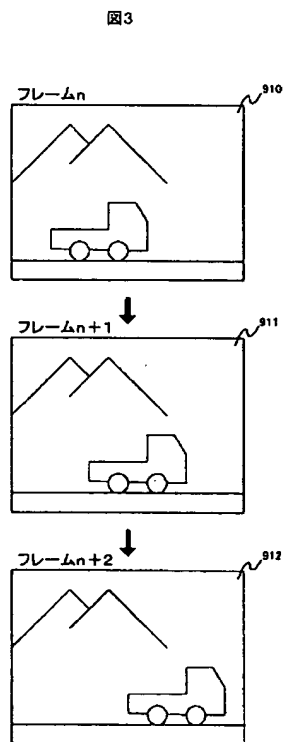
【図 1】



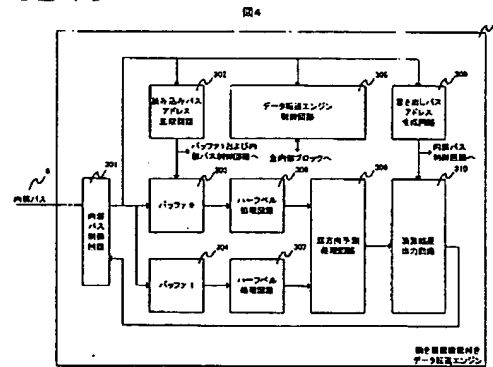
【図 2】



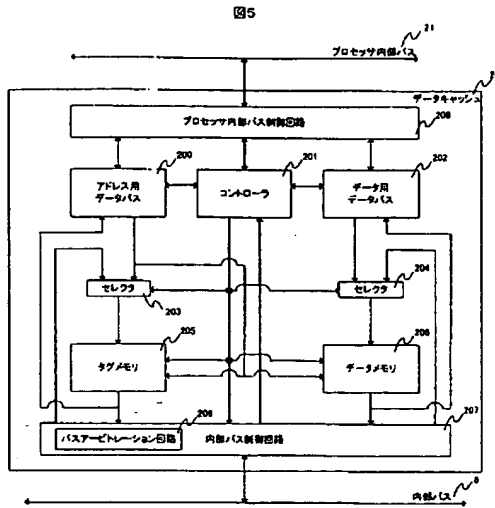
【図 3】



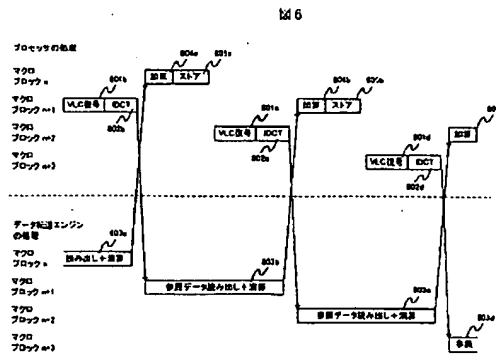
【図 4】



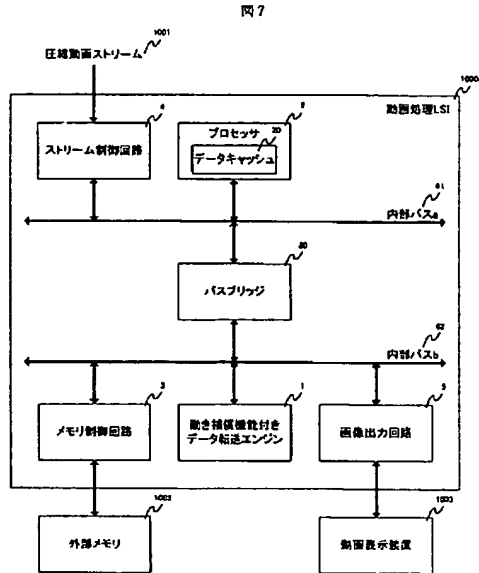
【図 5】



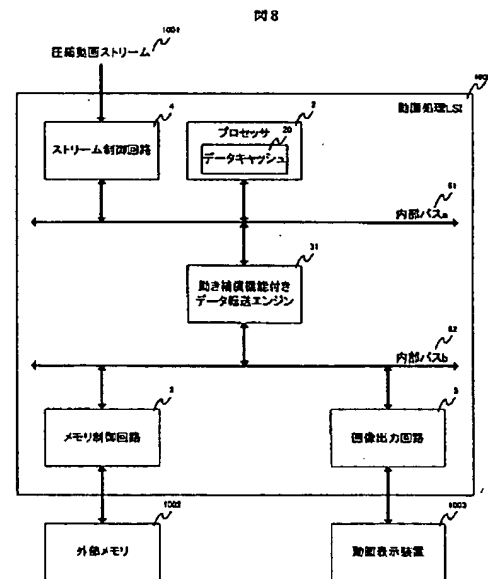
【図 6】



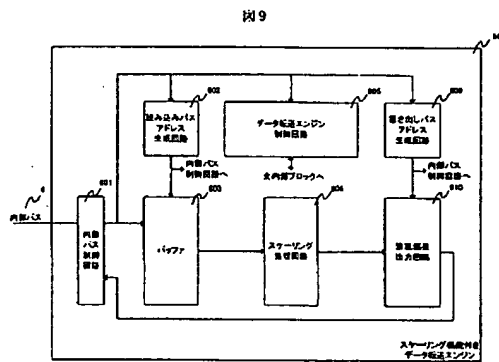
【図 7】



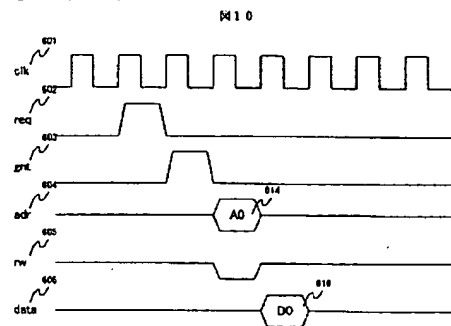
【図 8】



【図 9】



【図 10】



フロントページの続き

(51)Int. Cl.⁷

F I

テーマコード (参考)

G 0 6 T 1/20

G 0 6 F 15/16 6 4 5

5 C 0 5 9

G 0 6 T 7/20

G 0 6 F 15/78 5 1 0 G

5 L 0 9 6

H 0 4 N 7/24

G 0 6 T 1/20 ㄻ

G 0 6 T 7/20 B

H 0 4 N 7/13 ㄻ

(72)発明者 東嶋 重樹

神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

(72)発明者 西岡 清和

神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

Fターム(参考) 5B005 JJ12 KK13 LL11 MM01

5B013 DD03

5B045 BB12 BB28 BB29 DD01 DD12

5B057 CA12 CA16 CB12 CB16 CG02 CH04

5B062 CC04 CC05 DD01 FF05

5C059 KK50 MA00 MA05 MA31 NN01 PP04 SS26 UA05 UA33 UA36

UA38

5L096 HA04 LA01 LA13

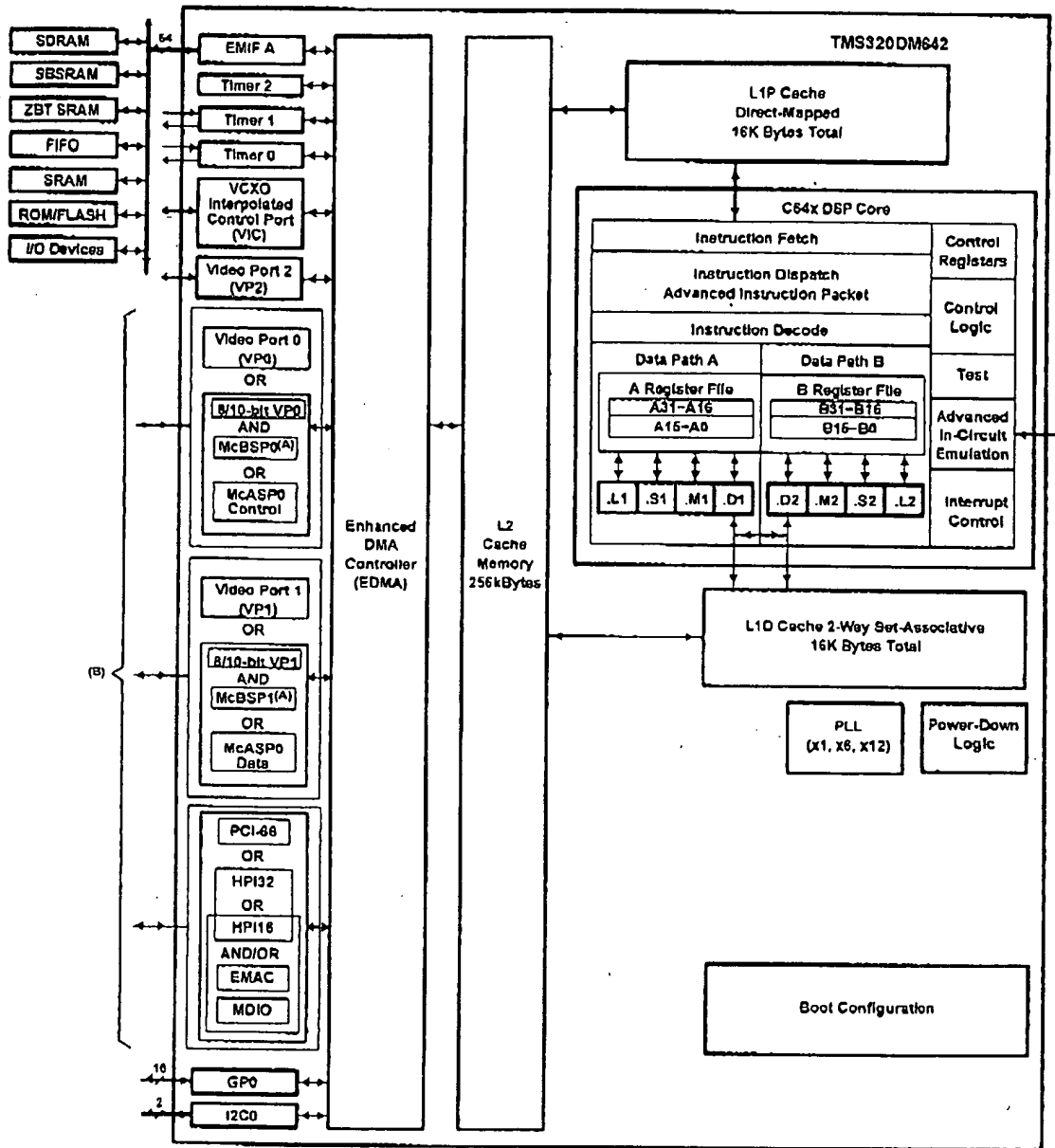
TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor

SPRS200K-JULY 2002-REVISED JULY 2006



1.3 Functional Block Diagram

Figure 1-1 shows the functional block diagram of the DM642 device.



- A. McBSPs: Framing Chips – H.100, MVIP, SCSA, T1, E1; AC97 Devices; SPI Devices; Codecs
- B. The Video Port 0 (VP0) peripheral is muxed with the McBSP0 peripheral and the McASP0 control pins. The Video Port 1 (VP1) peripheral is muxed with the McBSP1 peripheral and the McASP0 data pins. The PCI peripheral is muxed with the HPI(32/16), EMAC, and MDIO peripherals. For more details on the multiplexed pins of these peripherals, see the Device Configurations section of this data sheet.

Figure 1-1. Functional Block Diagram